

SyncWare News

The journal for technical
applications of T/S
computers

Volume 2 Number 4

Mar.-Apr. '85

How Fast is Fast Enough?

Today's word is SPEED. Regardless of which machine you own, you can speed up your SAVE and LOAD times in several different ways. The most economical (?), is the tape based fastload, of which there are several programs out for the 1000. The 2068 has a speed limitation due to design inconsiderations, although it does load six times faster than the ZX81. Follow Part II of the Cassette Connection to find out how you can raise this speed limit. This same limitation holds true for Spectrum conversions also. (Stay tuned for Part III.)

Stringy Floppy

The next step up is the Stringy Floppy, a high speed tape based system, of which the A&J version is quite popular. There is a version for the 1000 and 2068 (no 2068 Spectrum version yet). This system provides an order of magnitude speed increase over conventional tape systems. I advise against using tapes over twenty feet in length (stability problems: tapes stretch) and never write new data over old data without reformatting to clear away the previous garbage. This system works well if you do not add on extra boards to your system or otherwise vary the overall load that the CPU sees. The 1000 version is much less reliable than the 2068 system (I have one). Another step up is the Microdrive. This requires a Spectrum Rom and an adaptor to work properly. Whether this is a cost effective alternative to a disk drive is questionable. The wafers are expensive and don't hold much.

Floppy Disk

Enter, the Disk Drive. Let's be realistic. Although the disk interface costs more initially, disk drives and disks are cheap. The drives are like monitors in that they can be hooked up to virtually any computer (your loss is in the interface). For the 1000, there are the Aerco and the Compusa interfaces. The Aerco is faster, but the Compusa is very user friendly. (This whole issue is stored on a single Compusa disk. I couldn't imagine trying to keep all of this straight on individual tapes or even a half a dozen wafers.) Ramex has their 2068 disk drive on the street now. I have not yet seen it, but I understand that it is quite easy to use. Aerco is still working on their system. It promises to be very versatile. John Olinger and Ray Kingsley have pooled their talents hardware and software wise to create still another disk based system. This should prove to be the most economical of the three and easiest to modify ("kit"ers take note). Expect a release in April.

This overview brings up an important point: When the Speedwagon comes by, jump on. Don't get left in the dust!

FOR YOUR SUPPORT

This column announces any software or hardware that is new or otherwise untested by us. If you have something of interest, send us a description of your product. We advise readers to send a SASE, as a courtesy, to the individuals or companies, for further information.

English Micro Connection, 15 Kilburn Ct., Newport, RI 02840, 401 849-3805, offers Spectrum+'s, microdrives, expansion systems and Spectrum programs. SASE for more details or call.

TEJ Computer Products, 859 N. Virgil Ave., Los Angeles, CA 90029, 213 665-5111, has Z-Link, which will allow running of the Microdrives and other Spectrum hardware on a 2068 with an Emulator. \$49.95

Macshak Software, 73-312 Ironwood St., Palm Desert, CA 92260, 619 568-2082, has two new programs for the 2068. Investcalc, is an investment management program. They have a 1000 version also. Casino Keno, other other entry, speaks for itself. \$19.95+\$2.00p&h each.

A&J Micro Drive is available from the following suppliers:

Knighted Computer, 707 Highland St., Fulton, NY 13069, 315 593-8219

Curry Computer, 5344 Banff, Glendale, AZ 85036, 602 978-2902

Games to Learn By, PO Box 78, Collinsville, CT 06022, 203 673-7089

Gladstone Electronics, 1736 Avenue Rd., Toronto, Ontario, Canada M5M 3Y7

E. Arthur Brown, 3404 Pawnee Dr., Alexandria, MN 56380, 612 762-8847

TS Services, PO Box 15214, Red Bank, TN 37415, 615 877-6328

Delphic Enterprises, 5618 Martinique Dr., Corpus Cristi, TX 78411, announces a 4K Toolkit for the 1000/1500. It has 16 utilities that are relocatable to wherever you can execute machine code. The Eprom version is \$25.00. The tape version is available for \$18.95 from Tom Woods.

Pleasantrees Programming, PO Box 2034, Mesa, AZ 85204, has a new catalog. They supply the A&J Micro Drive and will also have the Aerco Disk interface and CPM when it becomes available. Send for their catalog.

Communications Systems Center, Bridge St. Box 133, Hillsboro, NH 03244, 603 464-5733, has ZX81's and their individual parts available. Write for more information.

Toronto Software world, PO Box 84, Agincourt, Ontario, Canada M1S 3B4, has a new Hi-Res game for the ZX81. Rocket-man, is available for \$18.95 Canadian (about \$14.95 US).

T Dailey, 40 Aspen, Great Falls, MT 59405, has a word/file handling program written all in machine code for the 1000. It comes with many advanced features, such as, variable file length, auto update, search, etc., and can be accessed from Basic. \$10.00 on cassette, and \$12.50 for Aerco disk.

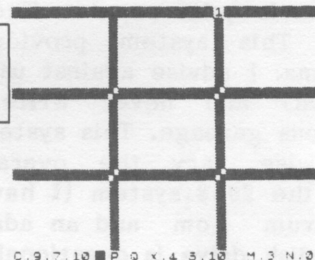
People's Software Supply, 609 E.N. 18th St., Abilene, TX 79601, 915 673-3538, has 2068 computers for \$145.95 and 2040 printers for \$65.95. (add \$5.00 for shipping.) They have Text Editor 2000, all in Basic for \$4.95 and the classic Star Trek, with Hi Res graphics, for \$4.95 also (add \$1.00p&h.)

S.J. Wyatt, PO Box 6239, Washington, D.C. 20015, has Kempston compatible joystick connectors that plug into the cartridge port. They will work with Romswitches and Roms (in 2068 and Spectrum mode) and with any other devices that do not use the cartridge port. \$8.00 bare board, and \$14.95 assembled and tested, \$1.00p&h.

John Oliger, 11601 Whidbey Dr., Cumberland, IN 46229, has a 2068 parallel printer port that is compatible with Aerco based software. It plugs into an expansion board. \$14.95 for the bare board, \$30.95 assembled and tested. It comes with a software cassette. John has printer cables available for \$16.95. Send for his catalog.

TIMESCREEN™
ZX81, 16K

Creativity and
planning aid:



- 3 screen formats (9-block format shown here)
- BASIC routines to enter and rearrange data
- Calendar
- 10 screens in 16K; 36/32K
- Use with any printer that will accept COPY routine

Cassette and manual, \$9.00 (VA res. add 4% tax), + \$1.00 s&h for one, 50¢ s&h ea. addl. cassette. One 8½x11 Label-Index Card sent with each cass.

HAWC™
Programming

4604 Apple Tree Drive
Alexandria, VA 22310

FORUM

Spectrum Compatibility

I'm sure that you are all aware by now that it is possible to run most Spectrum programs on a 2068. Some will run without modification and some will not. As indicated in the last issue (Games review, SWN 2/3), if a program stops due to an illegal RAMTOP (too low), then changing the CLEAR 24000 or so, to CLEAR 28000 or so, may allow the program to load and maybe execute as well. Self-contained programs (ones that make no specific routine ROM calls) will most probably run. If a lot of ROM calls are made, then they will not.

Enter, the Spectrum ROM. Now it becomes necessary to install a Romswitch, EMU or ROM into your computer. At this point, most (well written) programs for the Spectrum will run. Then, there are the rest. The true hacker on a ZX machine has been well groomed to write efficient code, compact subroutines and use all the tricks possible to speed up the program or otherwise make the program unique.

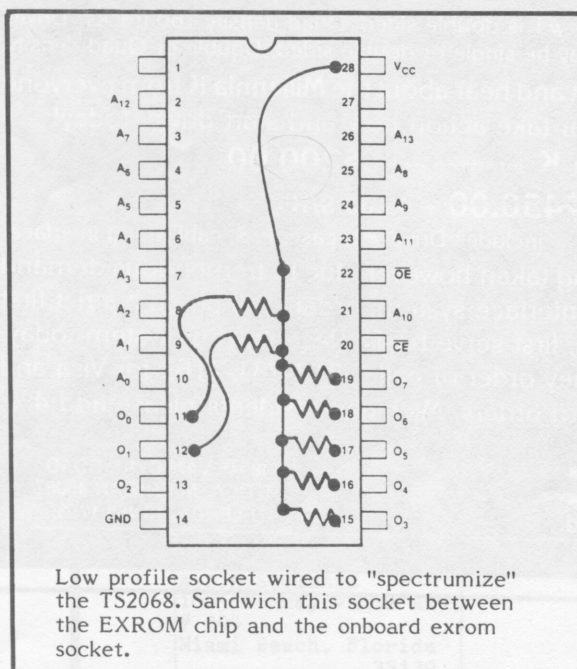
One such trick (and the one we can fix) is to assume that all Spectrums (in this case) act the same. Since the 2068 does not act like a Spectrum in its entirety; thus the problem emerges.

With the help of a lot of people and a lot of time invested, a partial solution has evolved.

First, the problem: The Spectrum (issue 3) puts an FF on the data bus (like the ZX81) after null keyscans or while it is just waiting with nothing to do. The 2068 does not.

The reason: All the data lines in the Spectrum have 10K ohm pullup resistors on them (like the ZX81). The 2068 does not. Only data line 2 is pulled high on the 2068.

The quick and dirty solution: Add 10K pullup resistors to the data lines.



The easiest place to make this modification such that if you screw up, you will not damage your machine in any way, is to use a 28 pin low profile socket and install it under the EXROM. Use the following pinout diagram to make the proper connections. You will need 7 resistors. I suggest using an in line resistor pack that has one common lead (not available from radio shack). This pin will be tied to +5 volts. This modification will not affect the running of any 2068 programs that I know of.

Remember, when you turn the socket over, you will have the mirror image of the diagram. It will therefore be necessary to solder on the opposite side as it appears here.

Hold the resistor pack so that all the pins are pointing up and the common pin is pointing away from you. Bend the first 5 pins (not the common pin) about 30 degrees to the left. Bend the rest of the pins (including the common pin in the other direction and all the way back over the side of the pack. Turn your socket over and use an alligator clip to hold the pack in place so that those 5 pins rest just on the lip of the socket over the data lines 3 to 7.

Remember, after you do the soldering, the socket must fit snugly, so don't use gobs of solder. After you tack down those 5 pins, you can remove the alligator clip.

Use 30 gauge wire wrap wire to jump D0 and D1 to two of the pins that you bent back.

Jumper the common pin to pin 28 of the socket. Open your case and locate the EXROM. It is the big chip above the cartridge bay. Pry it out and insert it into your "Spectrumized" socket.

Reinstall the circuit into the EXROM socket. Close up your case.

Testing the mod: The machine will run in the Spectrum mode whether your mod is correct or not. To test it, you must be in the 2068 mode. If the screen keeps reinitializing, then you goofed. If it comes up properly, then try booting one of those Spectrum programs (in the Spectrum mode) that previously did not work.

My thanks goes out to Bill Russell, John Oliger, John Brown, and Ray Kingsley.

Enjoy it!

BACKGAMMON

Looking for a real challenge? Tired of slow response time? Practicing for a tournament? If so, you'll want to try our Backgammon game. Features: full-screen graphics, break disabled, 100% machine code, clean loading, smart play, top rating from TSUG review, multi-game scoring, cube doubling, simple operation, documentation included. Requires ZX81 or TS1000 w/16K RAM. Available on cassette for only \$10. Purchase before 2/1/85 and receive our Z80 dis-assembler machine code monitor at no additional charge! Write for our FREE catalog listing 100 programs for TS 1000 and 2068. Please include exp. dt & phone # on MasterCard/VISA orders.

BIOCAL
SOFTWARE, INC.
167-C Wilson Street
Petaluma, CA. 94952

SPEED. REDEFINED

AN UNEQUALED EVOLUTION IN 2068 COMPUTING.

Imagine: you slide into your seat, select a program disk and slip it into your disk drive. In about the time it takes your finger to press the button, your selected program is up and running.

Realize: you don't have to imagine anymore—The new **Millennia K** disk interface and disk drive system from Ramex will enable your 2068 to operate in the league of computers costing 5 to 10 times as much! In fact, when it comes to maximum transfer rates, total storage capacity per disk and efficiency of formatted operation, the **Millennia K** will out perform Commodore, Apple IIe or IBM PC!

In order to extract the maximum efficiency from the disk, a whole new operating system had to be devised; the result: **SP-DOS**.

SP-DOS

offers these operating features:

- Loads 32K per second!
- Stores 820K per disk
- Occupies only 4K ram
- Compatible with most software

Super support!—a whole new generation of Disk Software:

- Tasword II (Word Processing)
- Tasmerge (Mailmerge)
- Masterfile (Database)
- Omnicalc II (Spreadsheet)
- Devpac (Assembler)

Language Support: • Basic
• Disk Pascal • "C" Language Compiler

The **SP-DOS** system allows you to use all the Timex cassette commands such as: • Save " " • Load " " • Save Data " "

- Load Data " " • Save Screen\$ " " • Load Screen\$ " "
- And all the Micro Drive Commands

Drives that may be used include

Qume, Tandon, MPI, Teac, Shugart etc.

Drive #1 is double sided, Quad density, 160 tracks. Drives #2,3,4 may be single or double sided, Double or Quad density.

You can wait and hear about the **Millennia K** from everyone else, or take action now and start drivin' today!

The **Millennia K** interface only: **\$200.00**

The Package: **\$450.00 + \$5.00 S/H**

includes Drive #1, case, P/S, cables and interface.

Orders are now being taken however, due to tremendous demand for the **Millennia K** interface system, orders will be on a strict first come, first serve basis. Reserve your system today.

Send check or money order or call 1-313-731-2163 for Visa and MasterCard orders. Michigan residents please add 4%.

RamexTM
International Ltd.

48945 VAN DYKE ROAD
UTICA, MICHIGAN 48087
TELE: 01-313-463-1795

Computer Error!

It has come to my attention that 15 people will not receive their issue of SyncWare News this time. If you know of someone who subscribes, but did not get their issue, then tell them to contact Tom Woods. They must be able to say when they subscribed. We know when they did, but we just don't know their names.

TYD-BYTS

Poking MODE

contributed by:
P.H. Skipper

The 2068 User Manual indicates that the system variable MODE, at address 23617, specifies the cursor (K, L, C, G, or E), and that "Poking the variable will have no lasting effect." This is not entirely correct.

First, the contents of MODE are determined as follows:

CONTENTS	BIT SET	CURSOR(S)
0	NONE	K, L, or C
1	0	E
2	1	G

If the system variable MODE is zero, then the cursor (L or C) is determined by the contents of the variable FLAGS2, at the address 23658. 0 (no bit set) will give the L cursor and 8 (bit 3 set) will give a C cursor.

POKE'ing the desired value into MODE will give the desired effect long enough to provide a preselected cursor in a following INPUT statement. This permits, for example, the following BASIC code:

```
10 BORDER 0: INK 6: PAPER 0: C
LS
20 INPUT "Location- Vert: ";v
   "      Horz: ";h
30 INPUT "Paper: ";p; "Ink: "
40 POKE 23617,2
50 INPUT "Graphic: ";g#
60 POKE 23617,0
70 PRINT AT v,h; PAPER p; INK
i;g#
75 PAUSE 200
80 GO TO 10
```

HELP for TASWORD

Duncan Teague

To modify the "HELP" pages of Tasword, STOP, go to BASIC and type in the command mode, the following lines:

```
FOR i=0 TO 1535: POKE (33280+i
),PEEK (54784+i): NEXT i
```

Then GOTO 20 and select "y" to return to the text file. The first of two "HELP" pages is now in the text file, and it can be edited in the normal way. When editing is complete, reverse the addresses and poke the screen back up in memory.

To edit the second "help" page, type in the following line:

```
FOR i=0 TO 1535: POKE (33280+i
),PEEK (55320+i): NEXT i
```

Repeat but again reverse the addresses to POKE this screen back up into high memory.

The following POKE codes have been used to adapt the Tasword Word Processor to the Aerco parallel interface

57999	127
58001	103
58003	127
58014	219
58015	127

Special printer symbols with SYMBOL SHIFT in EXTENDED MODE:

Key	Symbol	Instead of
y	[	
u	]	
p	@	copyright
a	~	slanted quotes
s		solid bar
d	\	
f	{	
g	}	

Special printer symbols with SYMBOL SHIFT in the normal mode:

Key	Symbol	Instead of
h	^	up arrow
x	#	pound sign

A.F.R. SOFTWARE

PRESENTS:

THREE EXTRAORDINARY PROGRAMS FOR THE

"Timex-Sinclair"
Computer

"ZX/CALC"- CALCULATOR

"ZX/TEXT"- WORD PROCESSOR

"ZX/CALENDAR"- APPOINTMENT CALENDAR

FOR COMPLETE DETAILS
WRITE:

A.F.R. SOFTWARE
1605 Pennsylvania Ave.
204
Miami Beach, Florida
33139

GIBSON Data Systems

Announces

A full powered business program for the expanded T/S 1000, 1500, and new 2068:

The Bookkeeper

GIBSON Data Systems
9 Orchard Drive Durham, NH 03824

all prices include U.S. Shipping

MASTERCARD

VISA

- up to 900 journal entries
- carry forward
- 99 user definable account names
- check register maintenance
- print out for:
 - journal with comments
 - ledger by account, date or all
 - chart of accounts
 - income statements
 - balance sheets
- item search
- easy corrections and changes
- for 32k or 64k with TS 2040 printer

Send \$25.00 for: Comprehensive manual
Cassette

CLEAR THAT SCREEN!

By looking at the title of this article, you might well think, "Well, ol' Fred's finally found the CLS command on his ZX81/TS1000!" Actually, I've known about it for a while now. The purpose of this article is to present some alternatives to this command, and in this process maybe show some ways of handling your display file. Also, it might get the ball rolling on our display contest (details later).

How a computer handles its screen functions is a major factor in the overall "look" that its programs have. The "down-wipe" CLS that is on the ZX81 family is somewhat non-typical, and is almost a signature for this machine. More common are the "zap" (as on the 2068) and the "scroll-away" (some other machines). The "zap" is particularly useful for animated games, where you want to get screens cleared in a hurry. The "scroll-away" is a somewhat more aesthetic approach, especially for screens of text or numeric data, giving the impression of screens being rolled up (or down) out of view. For other applications you might consider wipes, such as the "up-wipe" or "center-wipe" demonstrated later on. Side or diagonal wipes and scroll-aways are also possible; I'll leave implementation of these up to you enterprising readers.

If you are "Basic-bound," you can easily implement the up scroll-away using the SCROLL command, as in the following subroutine:

```
1000 FOR I=1 TO 22
1010 SCROLL
1020 NEXT I
1030 RETURN
```

You can significantly improve this by adding

```
1015 PRINT AT 21,31;" ";AT 21,0;
```

This fix, first suggested by John Oliger, helps smooth out the erratic behavior of the SCROLL command by padding each line after scrolling. Your manual says that "... SCROLL ... can upset this [the display file format] by introducing short lines at the bottom." That comment is a candidate for the "understatement of the decade" award - the SCROLL command is easily the worst routine in the ROM. If you get Tom Bent's improved "super-ROM," you will have a SCROLL command which is actually useful. If not, use the routine presented in this article, and avoid the SCROLL command entirely; you will thereby avoid a lot of display-file headaches.

If you're into machine-code, you shouldn't have much trouble understanding how the various CLS routines work, and might have fun writing others. If you're not, don't worry; I'll try to give enough information so you can implement the routines even if you don't know how they work. A great way (the best way?) to learn machine code is to study how other peoples' programs work - the annotations should help.

Start with the usual line 1 REM, at least 289 characters long. Then, enter the machine-code portion of the demo program using an assembler, such as Hot Z. If you're using Hot Z, save the REM to tape, then load Hot Z and use it to load the REM into high memory; enter the hexcode (left column) into the REM, then save it back to tape. Exit Hot Z and load back the REM (now filled with machine code)

and add the BASIC. Or, you can use the "Big REM" option to write/save the machine code. Alternately, enter lines 9000-9120 and feed in the decimal values from Table 1. When you've added the rest of the BASIC, save to tape for back-up and RUN. Most of the routines are fully relocatable to other areas of memory, and you can use just the one or ones you need for a particular application. The only exceptions are the UPCL and DNCL routines, in which you only have to change the single CALL to the appropriate scroll routine.

Zap

Before you expanded your machine to 16K, this is how the CLS looked. On adding more memory, the ROM pads out each line to a full 32 spaces, and the CLS changes into a down-wipe. The "zap" routine is actually a down-wipe too, but it is much quicker so the screen simply seems to vanish. It just goes through and fills each location in the display file with a zero (space). You can use other fill characters for a fast screen-fill; just POKE 16525 with the desired character code. For a "flash-zap," POKE it with 128 (inverse space), call the routine, POKE it back to zero, and call it again. You can also change how many lines are cleared by POKEing 16518 with one greater than the number of lines to clear. Unlike the rest of the routines here, the "zap" doesn't reset the PRINT position, though you can have it do this by adding a LD BC,16cc to specify line (1l; from the bottom up, including 2 for lines 22 and 23, which are line 0 and line 1 in the PRINT AT routine) and column (cc), then calling 08F5 in the ROM (set PRINT AT) before RETURNing.

Scroll Away

Many machines use the upward scroll-away for clearing the screen. It is also possible to have a downward scroll-away. For this purpose, two subroutines are provided: UPSC functionally replaces the existing SCROLL command; and DNSC, which is similar, but it scrolls down and resets the print position to the top instead of the bottom line. The UPCL and DNCL routines simply call the appropriate scroll routine 22 times for scroll-away CLS. UPSC actually transfers the contents of line 23 into line 21, so if you PRINT something at line 23 after POKEing 16418 with zero, and then call UPSC, this line will become your new bottom line. You might want to modify the end routine to set PRINT AT to line 23. You can then print the most recent data here, and have a stationary message in line 22. Similarly, UPCL can be used for a fast screen-fill of copyright notices and the like. The down-scroll uses a slightly different approach, and lets you choose the fill character with a POKE to 16613.

Up & Down Wipes

There is no point in writing a down-wipe routine, since the normal CLS does that admirably; however, an up-wipe is an interesting alternative, as the UPWP routine demonstrates. To keep things from going too fast and being just another "zap" routine, this one uses two nested loops to pause briefly between lines. Again, the fill character need not be a space (zero), so you can up-wipe a black screen or any other character you like.

WHILE THEY LAST...
BRAND NEW **TOP** QUALITY
DOUBLE SIDE DOUBLE DENSITY
DISC DRIVES \$ 99

2/3 HEIGHT 400 KILOBYTE CREAM COLORED
DUAL DRIVE CABINET, TOP QUALITY
HEAVY GAUGE STEEL WITH 5 AMP
SWITCHING POWER CONVERTER **\$ 99**

We want to say *THANK YOU* for the many months of your patience. Our Disc Interface for the TS2068 has 64K of on-board RAM which may be used as a second bank of system memory or a full-blown CPM system. It controls 4 floppy disc drives of any size- from 3 to 8 inches. 1 or 2 sides, single, double, or quad density (8" DD not yet supported). It also includes a complete RGB interface with separated sync so you can *SEE* the High Resolution Color Graphics that this machine produces. It does not include serial or parallel interfaces... you add as many as you like at \$ 69 for 1 Centronics Type Parallel Interface, P/N CP68 \$ 99 for dual independent RS-232 Serial Interface, RS68. The Disc Board is called FD68 and costs \$ 199. Deduct \$ 10 for the first CP68 or RS68. Add \$ 3 / item shipping.

We are now taking orders.



Box 18093, Austin, TX 78760
ACME ELECTRIC ROBOT CO. (512) 451-5874

Center Wipes

This one is the most complex of the routines. It clears the screen from the edges toward the center. It is a prime candidate for games and graphics programs, especially if the fill character is an inverse space. It works by spiralling around the screen 16 times, each time making a smaller square. It actually "clears" a square screen four lines larger at the top and bottom, keeping it symmetrical

as it approaches the center, and the conditional jumps to SKP1, SKP2, etc., check to make sure it's actually on-screen, otherwise it skips the load step to avoid overwriting program or variables. Register A is used as a step counter, for comparisons and to hold the upper limit of the vertical legs; B is the loop (spiral) counter and C holds the fill character. D holds the lower limit of the vertical legs, and DE is also used for the line offset (33 decimal). HL is the address pointer.

Table 1. CLS decimal listing
Enter each line of numbers
working down. Ex: 42, 12, 64,
6, 24, 126, 254, 118, 40, etc.

042:012:064:006:024:	032:000:209:237:176:	205:245:008:201:061:	001:000:016:017:131:	001:113:241:061:040:
126:254:118:040:005:	001:000:021:205:245:	062:022:245:205:203:	000:042:012:064:175:	003:043:024:242:062:
054:000:035:024:246:	008:201:000:062:022:	064:241:061:200:024:	237:082:017:033:000:	017:144:087:120:198:
016:251:201:061:042:	245:205:149:064:241:	247:061:042:012:064:	120:128:245:120:254:	016:254:030:048:005:
012:064:035:229:017:	061:040:002:024:246:	017:248:002:025:062:	013:048:001:113:241:	254:006:056:001:113:
033:000:025:209:001:	001:000:000:205:245:	024:245:054:000:001:	061:040:003:035:024:	061:186:040:011:245:
181:002:237:176:213:	008:201:061:042:012:	031:000:084:093:019:	242:062:017:144:087:	175:213:022:000:237:
017:033:000:025:001:	002:025:229:017:033:	237:176:062:003:006:	120:198:016:254:029:	082:209:241:024:232:
	000:025:229:209:225:	255:016:254:061:032:	048:005:254:005:056:	017:034:000:025:016:
	001:181:002:237:184:	249:017:064:000:237:	001:113:061:186:040:	157:001:000:000:205:
	225:006:032:054:000:	082:241:061:040:002:	007:213:022:000:025:	
	035:016:251:014:000:	024:225:001:000:000:	209:024:236:120:128:	
		205:245:008:201:000:	245:120:254:014:048:	

BASIC LISTING

```

1 REM EERND: 7/ PLOT ( CLS T
AN EERND7 FAST 15 SGN GOSU
B STR$ ) 5 4 SGN GOSUB +LN
PRINT TAN Y- PRINT LN RAND LET
XC / PLOT LN PRINT TAN EER
ND7 FAST 15 FAST SGN
LPRINT GOSUB LPRINT 40 7
( CLS LN PRINT TAN Y- PRINT
LN ACS RND LET XCOS / RUN EERND
) SAVE Y/ PRINT 0 3 << GOSUB
Y COPY ( RETURN X4 RAND ) RND
GOSUB 2 LET XC / LPRINT LN
PRINT TAN ( ) EERND GOSUB ?
5 2 PRINT ? RETURN K? LET XC
7/ PAUSE Y) LEN ( RETURN K)
RETURN 3 XUC STR$ - SGN / GOT
0 ? PRINT ? RETURN K? LET XC
F/ PAUSE Y) LEN ( RETURN 2K) R
ETURN 5 XUC PRINT STR$ - G
SUB ?SGN LET / CONT 06 ( LN
PRINT TAN
2 REM LINE 1= 289 CHAR. REM
3 SLOW
10 PRINT AT 0,0: " CLEAR SCRE
EN DEMONSTRATION "
20 PRINT "1. ZAP", "2. FLASH
ZAP", "3. UP SCROLL", "4. DOWN
SCROLL", "5. UP WIPE", "6. DOWN
WIPE", "7. CENTER WIPE", "8. FLA
SH WIPE", "9. STAR WARS", "0.
STOP "
30 LET E$="
35 LET Z$=INKEY$
40 IF Z$="0" THEN STOP
45 IF Z$<"1" OR Z$>"9" THEN GO
TO 35
60 GOSUB 1000
80 GOSUB 100*VAL Z$
90 RUN
100 REM ZAP =USR 16514
105 REM FILL CHAR. = POKE 16525
110 REM NO. LINES+1 = POKE 16518
120 POKE 16525,0
130 LET L=USR 16514
140 RETURN
200 REM FLASH ZAP
210 POKE 16525,128
220 LET L=USR 16514
230 GOTO 120
300 REM UPSCROLL
310 REM SCROLL COMMAND=USR 16533
320 REM UP SCROLL CLS=USR 16567
330 POKE 16418,0
340 PRINT AT 23,0: "NOT COPYRIG
HT BY FRED NACHBAUR"
345 LET L=USR 16567
350 PRINT AT 23,0: E$
355 IF INKEY$="" THEN GOTO 355
360 PRINT AT 22,0: E$
370 POKE 16418,2
380 LET L=USR 16567
390 RETURN

```

```

400 REM DOWNSCROLL
405 REM DN SCROLL COMMAND=16537
406 REM FILL CHAR. = POKE 16513
407 REM DN SCROLL CLS=USR 16524
410 LET L=USR 16524
500 REM UP-WIPE =USR 16535
505 REM FILL CHAR. = POKE 16547
510 LET L=USR 16535
520 RETURN
600 REM DOWN-WIPE (NORMAL CLS )
610 CLS
620 RETURN
700 REM CENTER-WIPE =USR 16584
705 REM FILL CHAR. = POKE 16585
710 POKE 16585,0
720 LET L=USR 16584
730 RETURN
800 REM FLASH-WIPE
810 POKE 16585,128
820 LET L=USR 16584
830 GOTO 710
900 REM STAR WARS
905 LET A$="*****"
906 REM INV.SPACE:INV.PERIOD:
GR.ON T:GR.ON G:INV.ASTERISK:
ASTERISK:PERIOD - OR ANY OTHER
PRINTABLE CHARACTERS
910 FOR A=1 TO LEN A$
920 POKE 16585,CODE A$(A)
930 LET L=USR 16584
940 POKE 16525,CODE A$(LEN A$-A
+1)
950 LET L=USR 16514
960 POKE 16585,0
970 LET L=USR 16584
980 NEXT A
985 GOSUB 300
990 RETURN
1000 PRINT AT 0,0:
1010 LET F$="SCREEN FILL****SCRE
EN FILL**"
1020 FOR A=1 TO 23
1030 PRINT F$:
1040 NEXT A
1045 POKE 16418,0
1050 PRINT F$(1 TO 14):TAB 5:"ANY
KEY TO CLEAR SCREEN"
1055 POKE 16418,2
1060 IF INKEY$="" THEN GOTO 1060
1070 RETURN
9000 REM DECIMAL LOADER
9005 FAST
9010 FOR A=16514 TO 16502
9020 IF A<>16590 THEN GOTO 9090
9030 POKE 16418,0
9040 PRINT AT 23,0:"ANY KEY, ENT
ER 9999 TO CONTINUE"
9050 POKE 16418,2
9055 PAUSE 4E4
9060 INPUT C
9070 IF C<>9999 THEN GOTO 9030
9080 CLS
9090 INPUT B
9100 POKE A,B
9110 PRINT (STR$ (1000+PEEK A)) (
2 TO )";";
9120 NEXT A

```


CLEAR SCREEN MACHINE CODE

HEXCODE	NAME	MNEMONIC	
4082 2A0C40	ZAPP	LD HL, (DFIL)	--ZAP ROUTINE--
4085 0619		LD B, 19	:get display file start address
4087 7E	NXCH	LD A, (HL)	:Number of lines + 1
4088 FE76		CP 76	:is it end of line?
408A 2805		JR Z NEXL	
408C 3680		LD (HL), 80	:yes- go on to next line
408E 23	INCR	INC HL	:no- fill location with fill character
408F 18F6		JR NXCH	:increment location
4091 10FB	NEXL	DJNZ INCR	:next character
4093 C9		RET	:decrement line, loop back if not zero
4094 00		NOP	:if done
4095 2A0C40	UPSC	LD HL, (DFIL)	--UPSCROLL ROUTINE--
4098 23		INC HL	:get display file start
4099 E5		PUSH HL	:first character on top line
409A 112100		LD DE, 0021	:save it
409D 19		ADD HL, DE	:line displacement = 33 decimal
409E D1		POP DE	:line 2 start address in HL
409F 01B502		LD BC, 02B5	:get line 1 start address in DE
40A2 EDB0		LDIR	:number of bytes in BC = 21 lines
40A4 D5		PUSH DE	:transfer 21 lines upward
40A5 112100		LD DE, 0021	:DE at end of line 20 - save it
40A8 19		ADD HL, DE	:line displacement
40A9 012000		LD BC, 0020	:HL had end of line 21; add a line
40AC D1		POP DE	:number of bytes = 32 decimal
40AD EDB0		LDIR	:destination = line 21
40AF 010015		LD BC, 1500	:transfer line 23 to line 21
40B2 CDF508		CALL PRAT	:print position at line 21, col. 0
40B5 C9		RET	:set print position
40B6 00		NOP	:return
40B7 3E16	UPCL	LD A, 16	--UP-SCROLL CLEAR SCREEN ROUTINE--
40B9 F5	NEXT	PUSH AF	:initialize counter at 22
40BA C09540		CALL UPSC	:save it
40BD F1		POP AF	:scroll up one line
40BE 30		DEC A	:get counter
40BF 2302		JR Z END0	:decrement it
40C1 18F6		JR NEXT	:done? exit if so
40C3 010000	END0	LD BC, 0000	:no - loop back
40C6 CDF508		CALL PRAT	:print position at top of screen
40C9 C9		RET	:set it
40CA 00		NOP	:return
40CB 2A0C40	DNSC	LD HL, (DFIL)	--DOWN-SCROLL ROUTINE--
40CE 23		INC HL	:display file start
40CF E5		PUSH HL	:line 0, column 0
40D0 11B302		LD DE, 02B3	:save it
40D3 19		ADD HL, DE	:distance to end of line 20
40D4 E5		PUSH HL	:get end line 20 into HL
40D5 112100		LD DE, 0021	:save it
40D8 19		ADD HL, DE	:line displacement
40D9 E5		PUSH HL	:HL now has end of line 21
40DA D1		POP DE	:save it
40DB E1		POP HL	:and get it back in DE
40DC 01B502		LD BC, 02B5	:end of line 20 in HL
40DF EDB8		LDLR	:number of bytes = 20 lines
40E1 E1		POP HL	:transfer downward
40E2 0620		LD B, 20	:get line 0, col. 0 address
40E4 3600	FILL	LD (HL), 00	:B is our space counter =32 dec.
40E6 23		INC HL	:fill it with fill character
40E7 10FB		DJNZ FILL	:next location on line 0
40E9 0E00		LD C, 00	:loop back until line 0 filled
40EB CDF508		CALL PRAT	:reset column to 0
40EE C9		RET	:and set print position before
40EF 00		NOP	:returning
40F0 3E16	DNCL	LD A, 16	--DOWN-SCROLL CLEAR SCREEN ROUTINE--
40F2 F5	NXLN	PUSH AF	:initialize counter
40F3 C0CB40		CALL DNSC	:save it
40F6 F1		POP AF	:down scroll one line
40F7 30		DEC A	:get counter
40F8 C8		RET Z	:decrement it
40F9 18F7		JR NXLN	:done? return if so
40FB 00		NOP	:no - loop back
40FC 2A0C40	UPWP	LD HL, (DFIL)	--UP-WIPE ROUTINE--
40FF 11F802		LD DE, 02F8	:display file start in HL
4102 19		ADD HL, DE	:distance to start of line 23
4103 3E16		LD A, 16	:start line 23 in HL
4105 F5	NXLI	PUSH AF	:initialize line counter
4106 3600		LD (HL), 00	:and save it
4108 011F00		LD BC, 001F	:put fill character into column 0
410B 54		LD D, H	:number of bytes still left to clear=31
410C 5D		LD E, L	:start of line address
410D 13		INC DE	:gets moved into DE
410E EDB0		LDIR	:increment to column 1
4110 3E03	PAUS	LD A, 03	:and clear the line
4112 06FF	LP01	LD B, FF	:initialize counter pause loop 1
4114 10FE	LP02	DJNZ LP02	:initialize counter pause loop 2
4116 3D		DEC A	:loop 2
4117 20F9		JR NZ LP01	:next loop 1
4119 114000		LD DE, 0040	:done? loop back if not
411C ED52		SBC HL, DE	:displacement to start of previous line
411E F1		POP AF	:back up
411F 3D		DEC A	:get counter
4120 2802		JR Z END1	:and decrement it
4122 18E1		JR NXLI	:exit if done
4124 010000	END1	LD BC, 0000	:otherwise loop back
4127 CDF508		CALL PRAT	:print position at line 0, column 0
412A C9		RET	:and set it
412B 00		NOP	:then return

```

4120 010010 CTWP LD BC,1000
4121 118300 LD DE,0083
4122 2A0C40 LD HL,(DFIL)
4123 AF XOR A
4124 ED52 SBC HL,DE
4125 112100 TOPL LD DE,0021
4126 78 LD A,B
4127 80 ADD A,B
4128 78 NXC1 PUSH AF
4129 78 LD A,B
4130 FE0D CP 0D
4131 3001 JR NC SKP1
4132 71 LD (HL),C
4133 F1 SKP1 POP AF
4134 3D DEC A
4135 2803 JR Z RITE
4136 23 INC HL
4137 18F2 JR NXC1
4138 3E11 RITE LD A,11
4139 90 SUB B
4140 73 LD D,A
4141 73 LD A,B
4142 0810 ADD A,10
4143 FE1D NXC2 CP 1D
4144 3005 JR NC SKP2
4145 FE05 CP 05
4146 3801 JR C SKP2
4147 71 LD (HL),C
4148 3D SKP2 DEC A
4149 BA CP D
4150 2807 JR Z BOTL
4151 D5 PUSH DE
4152 1600 LD D,00
4153 19 ADD HL,DE
4154 D1 POP DE
4155 18EC JR NXC2
4156 78 BOTL LD A,B
4157 80 ADD A,B
4158 78 NXC3 PUSH AF
4159 78 LD A,B
4160 FE0E CP 0E
4161 3001 JR NC SKP3
4162 71 LD (HL),C
4163 F1 SKP3 POP AF
4164 3D DEC A
4165 2803 JR Z LEFT
4166 28 DEC HL
4167 18F2 JR NXC3
4168 3E11 LEFT LD A,11
4169 90 SUB B
4170 57 LD D,A
4171 78 LD A,B
4172 0810 ADD A,10
4173 FE1E NXC4 CP 1E
4174 3005 JR NC SKP4
4175 FE05 CP 05
4176 3801 JR C SKP4
4177 71 LD (HL),C
4178 3D SKP4 DEC A
4179 BA CP D
4180 2808 JR Z NXLP
4181 F5 PUSH AF
4182 AF XOR A
4183 D5 PUSH DE
4184 1600 LD D,00
4185 ED52 SBC HL,DE
4186 D1 POP DE
4187 F1 POP AF
4188 18E8 JR NXC4
4189 112200 NXLP LD DE,0022
4190 19 ADD HL,DE
4191 109D DJNZ TOPL
4192 010000 LD BC,0000
4193 CDF508 CALL PRAT
4194 C9 RET
4195 00 NOP
4196 00 NOP
4197 00 NOP

```

```

--CENTER-WIPE ROUTINE--
:B=spiral counter, C=fill character
:displacement four lines back
:display file start
:clear carry
:get HL four lines before DFIL start
:line displacement = 33 decimal
:counter in A, then double it
:for number of spaces to fill
:save counter
:off-screen?

:skip it if so
:otherwise fill it with fill character
:get counter
:and decrement it
:continue if done with top leg
:else increment address
:and loop back.
:put 17 decimal into A
:and subtract spiral counter
:for lower limit in D
:then, put spiral counter in A
:and add 16 dec. for upper limit.
:over the top of the screen?
:if so, skip it.
:off the bottom?
:skip it if so.
:fill the location with character
:decrement counter
:is it at lower limit?
:go on to bottom leg if so.
:else save lower limit,
:and set displacement
:and move pointer to next line.
:get lower limit,
:and loop back.
:number of spaces = two times
:contents of spiral counter.
:save space counter.
:off the screen?

:skip it if so.
:else fill the location.
:get space counter
:and decrement it.
:all done? go on to left leg if so
:else decrement pointer
:and loop back.
:calculate
:lower limit
:and put it into D.
:upper limit
:into A.
:off the top?
:skip it if so.
:out the bottom?
:skip it if so.
:else load with fill character.
:decrement space counter
:and compare with lower limit.
:go on to next loop if done.
:save the space counter
:so we can clear carry
:and save lower limit
:so we can put line disp. in DE.
:back up one line,
:then reclaim lower limit
:and space counter before
:looping back.
:34 decimal = 1 space over, 1 line down
:for start of new spiral
:decrement spiral counter, loop back
:unless finished; restore print
:position, set it,
:and return.

```

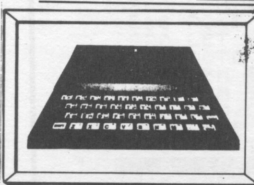
There. Wasn't that fun?



GET BACK TO BASICS with BASICALLY PROGRAMMING

basically
programming

2528 W. Olive
Fullerton, CA 92633
(714) 738-0666



TS2068 Programs
Multi-Draw 2068.....\$24.95
Softsync Assembler.....\$19.95
Voice Chess.....\$24.95
Personal Accountant.....\$24.95

TS1000 Programs
EZ-HEX.....\$12.95
Programmers Toolkit.....\$14.95
ZX Pro-File (16K to 64K).....\$16.95
ZX Data Finder.....\$14.95
Z-Wryter (16K to 64K).....\$12.95
Profit Plan.....\$12.95
Quest for the Holy Grail
& the Elusive Mr. Big.....\$17.95
3D Black Star.....\$ 9.95
Torpedo Attack.....\$ 9.95
Games Pack (2K).....\$14.95



Please include
check or money
order. Calif.
residents add
6% sales tax.
Please add
\$2.50 shipping
and handling.
No COD's - US
funds only.

SEND NOW FOR
FREE CATALOG!

BP - We have
not abandoned
basic computer
users!

SPECTRUM ROM SPECTRUM SOFTWARE

Considering interchanging your TS2068 ROM
with the Spectrum ROM?

We carry a comprehensive inventory of 48K
Spectrum software — arcade, adventure, busi-
ness and utilities for the TS2068/Spectrum.

Write for information and software brochure to

E. MCGHEE
Suite 150, 10127 - 121 Street
Edmonton, Alberta, Canada T5N 3X1

2068 WORD IMAGES

Thomas B. Woods

One particularly strong fascination the computer holds over me is its almost magical ability to conjure up powerful images in my mind. I'm not talking about the dazzling graphic TV images of an arcade game. These are nice, but far stronger are the pictures YOU create in YOUR mind in response to written words.

Have you ever read a good book that simply swept you away to some other time and place? Skilled writers know the power that words and language hold. With a few strokes of their mighty pens they wield enormous influence. Writers can cast their spells and wrap us around their little fingers. This magic comes from the words they use.

Surprisingly, most readers are unaware (and many will deny vehemently) that simple words can control our thoughts and consequently our behavior. The subtle nature of this mind control technique stems from our own imaginations. Whenever we look at written words, all we ever really see are symbols printed on paper. It is our minds which attach meaning and significance to the symbols.

Anyone, in possession of a daedal touch, may slip into your head at will! You can use the computer to illustrate this phenomenon by programming it to choose and print words on the TV screen. By consciously thinking about the images your mind creates in response you'll soon realize the powerful spell that words hold over you.

Enter this listing and RUN it. The program gives you the choice of determining your own sentence structure or letting the computer do it for you. The word types you can choose are 1. nouns, 2. verbs, 3. adjectives, 4. adverbs, and 5. prepositional phrases. To create a sentence consisting of an adjective, a noun, a verb, and finally a prepositional phrase, you would enter the command "3/1/2/5" when lines 60-100 ask you to input the sentence structure you wish. Remember to use the diagonal slash (/) between each number. Instead of entering a command at this point, you can also press just ENTER. This tells the computer to make a selection for you. It will randomly choose a structure from those programmed into lines 8000 to 8007.

Once the structure is finally decided upon, the computer forms sentences by taking the appropriate words from the DATA statements in lines 9000-9050. You can experiment with your own words by plugging them into these lines. Note, however, that this listing will READ only the first 20 words given in each DATA line. This is done by line 1015 which you can alter to suit your needs.

You can stop the generation of sentences to create a new structure by holding your finger on the space bar until the Selection menu comes up on the TV.

```

10 REM * WORD IMAGES *
15 LET S$=""
18 LET NOUN=9000
20 LET VERB=9010
22 LET ADJECTIVE=9020
24 LET ADVERB=9030
26 LET PREP=9040
28 LET OBJECT=9050
30 LET P=0
50 PRINT AT 3,7;"*** WORD IMAGES ***"
70 PRINT : PRINT "Select the sentence order you want to produce. EXAMPLE:"
80 PRINT : PRINT "3/1/2/4 (ENTER)": PRINT "prints a sentence order of ADJECTIVE, NOUN, VERB, ADVERB"
90 PRINT : PRINT TAB 1;"1-NOUN";TAB 1;"2-VERB";TAB 1;"3-ADJECTIVE";TAB 1;"4-ADVERB";TAB 1;"5-PREPOSITION"
95 PRINT AT 20,0;"Press ENTER for computer selections"
100 INPUT A$
102 IF A$="" THEN GO TO 8000
105 CLS
110 REM Set-up
115 RANDOMIZE
117 LET C=1
120 FOR X=C TO LEN A$: IF A$(X) <> "5" THEN NEXT X: GO TO 125
122 LET A$=A$(TO X)+"/6"+A$(X+1 TO )
125 FOR X=1 TO LEN A$
130 IF A$(X)="/" THEN NEXT X
140 IF A$(X)="1" OR A$(X)="2" OR A$(X)="3" OR A$(X)="4" OR A$(X)="5" OR A$(X)="6" THEN GO SUB 1000:VAL A$(X)
145 NEXT X
200 REM print sentence
202 PAUSE 300: CLS
205 FOR X=1 TO LEN S$
210 IF S$(X)="" THEN PRINT
215 PRINT S$(X);
220 NEXT X
225 IF INKEY$="" THEN RUN
230 LET S$="": IF P=0 THEN GO TO 125
235 GO TO 8001+INT (RND*7)
1000 REM noun
1010 RESTORE NOUN
1015 FOR Y=1 TO 20
1020 READ W$: IF INT (RND*10)=3 THEN LET S$=S$+" "+W$: RETURN
1025 NEXT Y
1030 GO TO 1001+VAL A$(X)
2000 REM verb
2010 RESTORE VERB
2020 GO TO 1015
3000 REM adjective
3010 RESTORE ADJECTIVE
3020 GO TO 1015
4000 REM adverb
4010 RESTORE ADVERB
4020 GO TO 1015
5000 REM preposition
5010 RESTORE PREP
5020 GO TO 1015
6000 LET S$=S$+" THE"
6010 RESTORE OBJECT
6020 GO TO 1015
8000 LET P=1
8001 LET A$="1/2/1/5/6": GO TO 125
8002 LET A$="3/1/5/6/4/2/1/5/6": GO TO 125
8003 LET A$="3/1/2/4": GO TO 125
8004 LET A$="1/2": GO TO 125
8005 LET A$="1/5/6/2/1": GO TO 125
8006 LET A$="1/5/6/2/4": GO TO 125
8007 LET A$="3/1/4/2/1/5/6": GO TO 125
9000 DATA "CHILDREN","PEOPLE","WORKERS","WARRIORS","WITCHES","LOVERS","BOYS","GIRLS","DOCTORS","BUTCHERS","PROGRAMMERS","ZOMBIES","MAGICIANS","TRUCKERS","ACTORS","POLITICIANS","MANIACS","PRIESTS","SAGES","ANALYSTS"

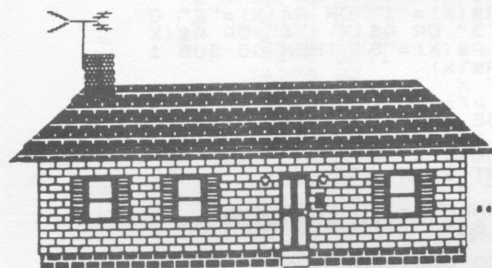
```

Line 225 scans the keyboard after each sentence is printed. If you're pressing the SPACE when this line is executed, then the computer will return to the start. Otherwise, a new sentence will be generated.

I guarantee that the sentences this program builds will trigger your imagination to produce all sorts of pictures and feelings. They'll range from philosophical universal truths to raunchy raw unexpurgated obscenity. If you become upset by what you see on the screen, don't blame me. Remember, the computer only prints symbols. It's YOUR imagination that's doing the dirty work.

I sat entranced for quite some time after I wrote this image maker. Some of the program's results were so suggestive that I dare not expose them in SyncWare News, a journal of such high moral character. But some of the others I particularly liked were, "Workers woo people in the sewer", "Truckers in the street quickly brainwash people in the office", and best of all, "Workers beneath the outhouse watch carefully". I don't want to spoil all the fun so I won't tell you any more. Just sit back, let WORD IMAGES cast its spell, and let your imagination soar! (What's a daedal look like?-ed.)

9010 DATA "ENCOURAGE", "PLUNDER",
"ATTRACT", "WOOL", "CARESS", "TOUCH",
"WATCH", "INSPIRE", "BRAINWASH",
AMUSE", "EVADE", "FIGHT", "SEDUCE",
"INSULT", "BITE", "EAT", "PECK", "AT
TACK", "EMULATE", "HONOR"
9020 DATA "BIG", "SMALL", "SWEET",
"SOUR", "PASSIONATE", "SILENT", "LO
UD", "INTIMATE", "FAT", "SKINNY", "S
AD", "JOYOUS", "HUNGRY", "THIRSTY",
"SATISFIED", "GREEDY", "GENEROUS",
"PEACEFUL", "RAGING", "STUFFED"
9030 DATA "QUESTIONINGLY", "SELF I
SHLY", "FEARFULLY", "BRAVELY", "AU
THORITATIVELY", "CARELESSLY", "CAR
EFULLY", "CONFIDENTLY", "HALFHEART
EDLY", "ENTHUSIASTICALLY", "STUPID
LY", "COLDLY", "WARMLY", "THOUGHTFU
LLY", "MADLY", "BLISSFULLY", "GRUDG
INGLY", "QUICKLY", "SLOWLY", "WISEL
Y"
9040 DATA "IN", "ABOUT", "AROUND",
"FROM", "CLEANING", "LEAVING", "BEN
EATH", "ABOVE", "PACING", "WATCHING",
"BESIDE", "NEAR", "AT", "WRECKING",
"BEHIND", "ABOUT", "ACROSS", "ALO
NG", "WITHIN", "CRITICIZING"
9050 DATA "HOUSE", "OFFICE", "STRE
ET", "SEWER", "YARD", "TUNNEL", "CAS
TLE", "MAZE", "DESERT", "SWAMP", "FO
RT", "FOREST", "HIDEOUT", "CITY", "B
EDROOM", "COFFIN", "OUTHOUSE", "PEN
THOUSE", "CAVE", "GRAVEYARD"



KNIGHTED COMPUTERS

"YOUR 2068 STORE"

MULTI-DRAW 2068

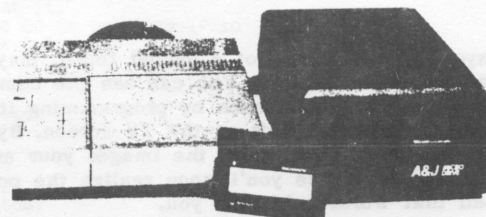
by Gary E. Ward

The two pictures on the left were made using MULTI-DRAW 2068. This program allows you to draw on the screen with only the use of the joystick! Turn on/off any pixel(s) at will (Instantly MAGNIFY the area in that your working on, for ease of use), then output your masterpiece to either a TS2040 printer, or a full size dot matrix type printer using either the AERCO or TASMAR interface. Also, with a full size printer attached, you have a choice of SMALL PRINT, LARGE PRINT, and GRAY PRINT (large size), and all can be accomplished by simply moving the the cursor with the stick to your desired choice. Some of the options available with the stick are:
DRAW COLOR BIG PRINT SMALL PRINT
GRAY PRINT MAGNIFY CLEAR SCREEN INK
PAPER SAVE SCREEN\$ LOAD SCREEN\$
SAVE PROGRAM LOAD PROGRAM BASIC

There are 28 predefined graphics characters (that you can quickly and easily re-define at will - again with only the use of the joystick. Many more features.

ONLY \$ 19.95 CASSETTE

(Expect to have available on Wafer by the time this ad is out - call us.



A&J MODEL 2000

MICRODRIVE

for your TS2068 - \$199.50

Transfer Rate = 11,400 Baud !!!

INCLUDES:

Microdrive and Interface
5 Microwafer II (1 ea. size)
Expansion Cartridge
Wafer Organizer - Wafer Wheel
Wafer Caddy

Owners Man. & 90 Day Warranty

Eliminate those LOAD/SAVE problems with this truly great system that approaches the speed of many of the current Disk Drives -- at half the cost !!! This system does not use any of the RAM in your computer like many other of the mass storage devices available do!

TS 2068 SOFTWARE ON CARTRIDGE

P I N B A L L \$ 21.95 FLIGHT SIMULATOR \$ 21.95

TS2068 CASSETTE SOFTWARE

STOCK MARKET SIMULATOR \$ 17.95 VU-3D \$ 17.00

Add \$3.00 to total order for shipping and handling

New York Residents add 7% Sales Tax

KNIGHTED COMPUTERS

315-593-8219

707 HIGHLAND ST.

FULTON, N.Y. 13069

Write for FREE Catalog

BASIL'S COMPENDIUM

John (Basil) Wentworth
1413 Elliston Drive
Bloomington, IN 47401

LET A= ...

This installment will teach you how to enter an actual program in machine code. But first, a "fun" program - a rather lengthy one, this time.

You'll notice that we have changed the loader program again, as shown in Figure 4-1. After the experience you've had in the first three chapters, you should be able to combine the operations of POKEing the vacancies and proofreading the program.

The program we're giving you this time will enable you to delete blocks of program material in a single operation. All you do is mark the beginning of the section to be deleted with

```
N1 REM >
```

and the end with

```
N2 REM <
```

where N1 and N2 are line numbers. Then RAND USR 16514 will lump all the material between the markers into one line which can be deleted at one go.

The rough draft of the 1 REM statement is shown in Figure 4-2, and the final listing in Figure 4-3. While you're proofreading the statement, the display will stop for you wherever you have included a period, and wait for you to INPUT the final number (which may, of course, actually be 27). When the screen is full, just CONTINUE to get the following bytes.

As before, the mnemonics, which are included for the information of more experienced programmers, will not appear on your listing.

One word of caution: DON'T use this program in the SLOW mode. It's very likely to crash if you do, since this program pre-empts some of the the computer's capabilities that are required to make the SLOW mode work.

Now back to lessons.

You've carved out some turf for yourself - so what do you do with it? You may remember an earlier declaration that the output of a machine

```
1000 LET F=16513
1010 LET F=F+1
1020 PRINT F;" ";
1030 IF PEEK F<>27 THEN GOTO 106
0
1040 INPUT A
1050 POKE F,A
1060 PRINT PEEK F,CHR$ PEEK F
1065 IF PEEK F=116 THEN STOP
1070 GOTO 1010
```

FIGURE 4-1. LOADER/PROOFREAD PROGRAM

```
1 REM <>5.RNDY.W<>7<> TAB
ND<> COS Y.<> TAB RNDY<> TA
B RND<> FAST AT W<> TAB RND<>
FAST LPRINT .PI777VAL <> LPRINT
<>.<> TAN
```

HOW IT LOOKS

```
1 REM <> 5 . RND Y . W <> 7
<> (SPACE) TAB (GR G) RND <>
(GR 1) COS Y . <> (GR 5) TAB
(GR Y) RND Y > <> (GR T) TAB
RND <> FAST AT W <> (GR T)
TAB (GR Y) RND <> FAST LPRINT .
PI 7 7 7 VAL <> LPRINT <> .
(GR 4) <> . (GR 7) TAN
```

HOW IT IS DONE

FIGURE 4-2. ROUGH DRAFT OF 1 REM

language routine is what the computer calls BC; i.e.

```
C+256*B
```

So how do we make B and C reflect any value that we can use?

For the moment, think of B and C as simple variables. What we want to do is to assign values to them. The Z80 machine code instructions that we want are:

```
LET B=m is 6,m
and LET C=n is 14,n
```

NEW LIFE FOR YOUR TS2068? YES! with IMPROVED ROMSWITCH

*Practically 100% Spectrum software compatible.

Lets you run Spectrum programs on your TS2068!

- * EASY INSTALLATION. No soldering, no drilling.
- * Just plug inside your TS2068 freeing edge connector & cartridge port for other uses.
- * External Switch-on switch selects Spectrum or TS2068 ROM

SPECIAL SALE--ROMSWITCH & 007 SPY--\$54.95
or ROMSWITCH & Winky Board 2000--\$69.90

007 SPY Tape Copier for SPECTRUM/TS2068 programs
with ROMSWITCH, Emulator, SPEC. ROM-cassette \$9.95

Send for free catalog of new hi-res programs for
TS1000, SPECTRUM programs & Manual, books etc.!

WINKY BOARD 2000 Cassette-computer interface for TS2068,
Spectrum, TS1000-1500, ZX81/80

- * Solves your LOADING problems
 - * Duplicates any TS/ZX cassette
 - * User friendly. Simply plugs into jacks.
- \$22.95 assembled/tested, shipping incl. U.S./Can.



RUSSELL ELECTRONICS

RD 1 * Box 539 * Centre Hall, PA 16828
814-364-1325 MasterCard/Visa 10am-8pm Check/MO

```

16514 221 <> LD IX,
16515 33 S 16509
16516 125 ?
16517 64 AND
16518 62 Y LD A,
16519 117 ? 117
16520 60 W INC A
16521 221 <> INC IX
16522 35 7
16523 221 <> CP (IX+0)
16524 190
16525 0
16526 194 TAB JP NZ,
16527 137 16521
16528 64 AND
16529 221 <> CP (IX+1)
16530 190
16531 1
16532 200 COS RET Z
16533 62 Y LD A,
16534 234 REM 234
16535 221 <> CP (IX+5)
16536 190
16537 5
16538 194 TAB JP NZ,
16539 134 16518
16540 64 AND
16541 62 Y LD A,
16542 18 18
16543 221 <> CP (IX+6)
16544 190
16545 6
16546 194 TAB JP NZ,
16547 168 16552
16548 64 AND
16549 221 <> PUSH IX
16550 229 FAST
16551 193 AT POP BC
16552 60 W INC A
16553 221 <> CP (IX+5)
16554 190
16555 6
16556 194 TAB JP NZ,
16557 134 16518
16558 64 AND
16559 221 <> PUSH IX
16560 229 FAST
16561 225 LPRINT POP HL
16562 237 GOSUB SBC HL,BC
16563 66 PI
16564 35 7 INC HL
16565 35 7 INC HL
16566 35 7 INC HL
16567 197 VAL PUSH BC
16568 221 <> POP IX
16569 225 LPRINT
16570 221 <> LD (IX+4),H
16571 116 ?
16572 4
16573 221 <> LD (IX+3),L
16574 117 ?
16575 3
16576 201 TAN RET

```

FIGURE 4-3. FINAL LISTING OF
1 REM

And RETURN, which we must not forget, is 201.

Note that the instructions (6,14) and the data (m,n) are mixed into the program. There is no difference between them, as far as mortal eye can see. The computer knows, though, from the context.

If it receives an instruction of 6 or 14, for instance, it'll interpret the next byte as data, whether you mean it to or not.

Now we're ready for our first, very simple, program. After reserving at least 5 nulls in a 1 REM statement (see the previous chapter, if you've forgotten how), POKE in the sequence shown in

```

MEANING
POKE 16514,6 LET B =
POKE 16515,1 1
POKE 16516,14 LET C =
POKE 16517,0 0
PRINT USR 16514 PRINT 256*B+C

```

FIGURE 4-4. SETTING B AND C

MEANING

```

POKE 16514,6 LET B =
POKE 16515,0 0
POKE 16516,14 LET C =
POKE 16517,1 1
PRINT USR 16514 PRINT 256*B+C

```

FIGURE 4-5. SETTING B AND C

Figure 4-4. Give the command PRINT (not RAND) USR 16514, and there's your answer: 256.

Now another one, as in Figure 4-5. Note that you don't have to delete the previous material - the new entries will overwrite the old bytes. Before printing the output, what do you expect it to be? Enter PRINT USR 16514, and check your prediction.

It should have been

$$256*B(=0)+C(=1)=1$$

But all this POKEing is getting to be a bit tedious, isn't it? (You'll notice that I stalwartly resist the temptation to make a pun on poking fun.) Plus the danger of losing your place

```

1 REM
1000 FOR M=16514 TO 17000 ***
1010 PRINT M;" ";
1020 INPUT A
1030 POKE M,A
1040 PRINT PEEK M
1050 NEXT M

```

```

*** DON/T WORRY; WE WON/T USE
ALL 487 BYTES. SEE THE TEXT
ON THIS POINT.

```

FIGURE 4-6. A SIMPLE LOADER

and skipping a byte. With the further disadvantage that you can't proofread the program.

It has probably already occurred to you that this is an appropriate place to use a loop. So let's build a very simple loader program, like the one in Figure 4-6.

To try it out, let's go through the two programs we wrote above. First, RUN the program of Figure 4-6, and then INPUT in succession:

6,1,14,0,201,Z

The Z, or any other letter that the computer will not interpret as a number (NOT M or A, which have already been assigned values by the computer), will cause the program to STOP, with an error report of 2/1120, meaning "I don't recognize that as a defined value for A". And that's the way you get out of the loop without having to go through all 487 INPUTs.

```

16514 6
16515 1
16516 14
16517 0
16518 201

```

FIGURE 4-7. PROOFREADING

Now you have the contents of the whole 1 REM statement displayed on the screen, as shown in Figure 4-7, so you can proofread it - including making sure that it includes the all-important 201, for RETURN. If it's the way you want it, PRINT USR 16514. And there's the 256, just as before.

If you do find an error in the program, POKE in the proper value, as, for instance, POKE 16516,14. If your program is to be more than 22 bytes long, use CONTINUE when the screen is filled. Or you can add 1005 SCROLL, if you prefer.

If you want to proofread a program after POKEing in a correction, use the following procedure:

1. Disable lines 1030 and 1040. You can do this either by deleting the lines or by inserting a REM at the beginning of each line, right after the line number.
2. If you have included a SCROLL, either disable it or put the computer into SLOW mode.
3. RUN the program.

El Minimo

The following experiment will help you to understand what goes on when you store your machine code in REM. It'll also give you a clue to the way we've been writing those "fun" programs at the head of each chapter.

I call this loading technique "EL MINIMO" because it's just about the most compact program loading system possible.

Enter

```
1 REM (Gr T) (space) : (Gr1) TAN
```

If you'll look at the listing of the character set in your manual, or if you'll check each character by PRINT CODE "...", you'll find that the codes of these respective characters are, to nobody's surprise, 6,0,14,1, and 201.

This is in fact the series of numbers that has been entered into memory, starting at 16514.

Try PRINT USR 16514 for this program, and again after you have exchanged the places of the 0 and the 1.

You may find it more convenient to use this technique for building up your 1 REM statement, as we have been doing in the "fun" programs. If so, you'll find it useful to add a little more information to your ZX81 manual. You'll notice that the table of the character set as listed in the manual shows the graphics symbols for Codes 1-10 and 128-138 (with an error in the entry for 135, at least in my copy). However, it's not too easy to find these symbols on the keyboard, so I wrote into the manual the keystroke needed to produce each of these symbols. They're listed in Figure 4-8.

Incidentally, if you want to enter keywords, like RUN (CHR\$ 247), REM (234), FOR (235), and the like directly from the keyboard, you can do it by preceding the character with THEN, which conditions

CODE	CHR\$	KEYSTROKE
0		SPACE
1	■	GR 1
2	■	GR 2
3	■	GR 7
4	■	GR 4
5	■	GR 5
6	■	GR T
7	■	GR E
8	■	GR A
9	■	GR D
10	■	GR S
128	■	INV SPACE
129	■	GR Q
130	■	GR W
131	■	GR 6
132	■	GR R
133	■	GR 8
134	■	GR Y
135	■	GR 3
136	■	GR H
137	■	GR G
138	■	GR F

FIGURE 4-8. KEYING THE GRAPHICS SYMBOLS

the ZX81 to interpret the next keystroke as a command. You'll have to delete the THENs from the final program, of course.

And that's it for now. There will be more.

2068 ART

David Kulp
CATS User Group
PO Box 725
Bladensburg, MD 20710

This short program will have you staring at your screen for hours. Type it in and let it RUN.

```
1 GO TO 200
60 FOR Z=1 TO 200
62 PLOT A,B: DRAW X-A,Y-B
70 LET A=A+A1: LET B=B+B1
80 LET X=X+X1: LET Y=Y+Y1
100 IF A>=255 OR A<=0 THEN LET
A1=-A1: LET A=A+A1
110 IF X>=255 OR X<=0 THEN LET
X1=-X1: LET X=X+X1
120 IF B>=175 OR B<=0 THEN LET
B1=-B1: LET B=B+B1
130 IF Y>=175 OR Y<=0 THEN LET
Y1=-Y1: LET Y=Y+Y1
135 NEXT Z
136 CLS
140 GO TO 60
199 REM ART
200 PAPER 0: INK 7: CLS
210 LET A=INT (RND*255): LET B=
INT (RND*175)
220 LET X=INT (RND*255): LET Y=
INT (RND*175)
230 LET A1=2: LET B1=2: LET X1=
4: LET Y1=4
240 GO TO 60
```

☆ MEMOTEXT ☆

IS NOW AVAILABLE ON TAPE

for your ZX81/TS1000/TS1500 and Memotech
Centronics or Serial Printer Interface

A FULL-FEATURE WORD-PROCESSOR AND FILE MANAGER
** USABLE WITH DISK SYSTEMS! **

By exclusive arrangement with Memotech, Inc., I
am able to supply this powerful program on
cassette tape. Look at these features!

- * Re-defined keyboard works just like a typewriter. Unshifted keys are lower case.
- * Supports separate data files to allow form letters, invoices, etc.
- * Comprehensive print-control commands plus direct hexcode control allows use of virtually every feature of your particular printer.
- * Multiple files may exist in memory at the same time. Only the files are saved/loaded, reducing time during cassette operations. Load/save files separately, or all at once.
- * Word-wrap and full-screen cursor makes editing a snap.
- * Comprehensive error checking, full search, exchange and move features.
- * Page numbering and headers supported.
- * Input "customizers" for form letters.

In addition to these features built into the program by Memotech, the tape versions include:

- * Powell Hargrave's "SDS" fast-save system saves/loads files about 10 times as fast as the normal tape routines. No need to quit Memotext; QUICK-SAVE, VERIFY and QUICK-LOAD are provided as additional functions.
- * QIT function allows you to quit Memotext and return to BASIC with ALL FILES INTACT! This allows use with disk systems as well as providing added flexibility.
- * Print routine sends a carriage return with every line-feed for printers requiring this.
- * Optional help file fully summarizes all functions, actions, and editing commands. Single key-stroke access from within any text file; single key return to exactly where you were editing.
- * Memory remaining report included with "display" function.

THREE VERSIONS AVAILABLE:

- ☐ V1 Uses memory from 8-16K as original module from Memotech. Requires memory in this range, e.g. Hunter board or 64K RAM. Allows up to 46K of text in 64K, or 14K in a 16K RAMpack. Includes 7-3/4K Help file and built-in "Functions" help screen. SPECIFY CENTRONICS OR RS232 I/F.
- ☐ V2 Located at 25-32K, allowing operation with a 16K RAMpack and printer interface alone. Usable with disk-based systems. Allows nearly 7K of text or data files. Help-file condensed to 2-1/2K (including "Functions" screen) to maximize available memory. All other features retained!
- ☐ V3 This version is the best choice for 64K owners who wish to use disk-systems such as Aerco or Compuas to store and retrieve Memotext files. It resides at 40-48K, and requires the "Oliger mod" (MI line decoding) on ZX81's and TS1000's. Frees up 8-16K range (except area used by printer I/F) for your DOS, while allowing over 22K of files. Help file resides at 48-56K, allowing same file space whether "help" is present or not. Otherwise identical to V1.

DON'T DELAY - ORDER NOW!!!

Price: US\$ 29.95 (\$37.50 Canadian) ppd. \$5.00 (\$6.25 Canadian) discount available if you already own the Memotext module; include a photocopy of your Memotech invoice. Additional versions - \$19.95 (24.95 Canadian).

ORDER FROM: Fred Nachbaur
Mountain Station Group Box 12
Nelson, BC V1L 5P1
CANADA

"SMART TEXT ZX/TS"

* INTERACTIVE SOFTWARE FOR *

TS-1500, TS-1000, ZX-81 WITH 32K RAM

FULL SIZE PRINTERS OR TS-2040

WORD PROCESSING, PRINTER CONTROLLING,
DATA BASE MANAGEMENT, MAILING LIST,

AND "INTERACTIVE OFFICE TOOLS".

"SMART TEXT ZX/TS" WILL FUNCTION WITH ALL MODELS OF ZX/TS USING THE 2040 PRINTER, ALL THAT HAVE THE "JOHN OLIGER PRINTER PORT" WITH OKIDATA ML-829 PRINTER. DESIGNED TO BE EASILY ADAPTABLE FOR ALL PRINTER INTERFACES, AND ALL FULL SIZE PRINTERS.

\$29.95 CHECK OR MD GETS THREE DIFFERENT VERSIONS OF SMART TEXT, AND 35 PAGE MANUAL. (SAVE WITH INQUIRIES)
WHEN ORDERING SPECIFY FOR EITHER "STANDARD ZX/TS", OR "OLIGER VDP".

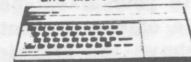
GULF MICRO ELECTRONICS, 1317 STRATFORD AVE, PANAMA CITY, FL 32404. (904) 871 4513

T-S Horizons

Each issue brings you 32 pages of usable information, program listings, product reviews, programming articles, hardware projects, and applications for your computer. T-S HORIZONS features nationally known authors like Paul Hunter, Gordon Young, and others. The price is ONLY \$15.00 for a full year (12 issues). AND for a limited time only, new subscribers will receive our special telecommunications issue (#7, see below) *FREE* with their subscription.



IN T-S HORIZONS #7
Byte Back Modem Review
2048 Programming Tips
How a Compiler Works
Telecommunications
for Beginners
Book & Game Reviews
Rotating Globe Routine
Articles By G. Young
and Paul Hunter
Sinclair Info. Network
TS-1000 Power Supply
-and more!!!



T-S HORIZONS

\$15 for 12 monthly issues. \$2.00 sample or back issues.
Over thirty pages of fun, fact, and great computer information.

Enclose \$15 (US) for 12-issue subscription, plus *FREE* back issue (while supplies last).
\$21 Canada, \$25 other foreign.
Extra back issue/sample: \$2.00

NAME

ADDRESS

CITY

STATE, ZIP

SEND TO: T-S HORIZONS
2002 Summit St
PORTSMOUTH, OH 45662

HARDWARE PROJECTS

Make Your Own 2068 & TS1000 EPROM Programmer

John Oliger
11601 Whidbey Drive
Cumberland, IN 46229

Here is a project that will let you "burn" (program) EPROMs using your computer. It will handle either the 2764 type (8K bytes) or the 27128 (16K) EPROMs. The text assumes that this is for the TS2068, as this is the machine for which the board was designed. (The circuit works quite nicely with the ZX81/TS1000 also.-ed.)

The 2068 EPROM Programmer is memory mapped at 8192-16383 in the 2764 (8K) mode, and at 0-16383 in the 27128 (16K) mode. The programmer will respond only to a write (POKE) in this address space. If the programmer is installed in the expansion board (or otherwise plugged on the back of the 2068, but the 1000 requires an expansion board due to the size of the PC Board), contains a fresh EPROM, and has 21 VDC applied to its Vpp input line, then it will take any data POKed into it and store it on EPROM at the address POKed for the 27128, or at (address POKed - 8192) for the 2764. The programmer has its own timer and latch circuit, and DOES NOT use the Z80 WAIT NOT control line, so it is compatible with the dynamic memory used in the 2068; there is no possibility of data corruption. Unlike most EPROM programmers, the software required to program an EPROM consists of a simple FOR/NEXT loop with a PAUSE 3 statement after each POKE in the loop to allow the timer on the board to complete its programming cycle. Due to the design of the programmer, data verification from this board is not possible. If verification is desired, it must be performed from a 2068 User Cartridge (procedures for doing so will be covered later). EPROMs with a rated access time of 300 ns or less are recommended for the 2068 computer, although this programmer is capable of programming any speed of EPROM. I should mention that I've experienced problems with speed using Intel brand EPROMs, and have read of the same problem with Intel EPROMs used on other computers (the BBC comes to mind immediately), so I recommend against using this brand on the 2068, if possible.

Buying EPROMS

One of the best places I know of to get the EPROMs for this programmer is Microprocessors Unlimited, phone (918) 267-4961. Current price for the 300 ns. Hitachi 27127 is \$12.97, and for 2764 it is \$5.00.

Static is a NO-NO!

Before we go on to the construction details, a few words about static electricity. STATIC IS THE #1 KILLER OF EPROMS. Yes, static can blow that \$12.00, 27128 in no time flat. There is no reason to become "static paranoid" about this though. I recommend the following two rules, which if strictly adhered to will prevent 99% of all static problems:

- 1: Don't even look at your computer or any EPROMs without first removing your shoes! Socks are OK, but NEVER wear shoes while working on or near your computer.
- 2: If you need to ship a board containing EPROMs or other static sensitive MOS parts to me or anyone else, wrap the entire board in aluminum foil. A lot of static can build up in that white packing material and other plastics.

Equipment

1. Use a clean, low-wattage, fine tip soldering pencil.
2. Get a very small diameter 60/40 rosin-core solder to assemble the board.
3. Q-tips and a bottle of acetone or (preferably) non-flammable commercial flux remover.
4. You will need a pair of flush cutting "nippy" cutters (Radio Shack is one source) for cutting component leads after soldering. You might possibly need a small pair of needle-nose pliers to form the resistor and cap leads before installation.

Construction

As usual, there are a number of different approaches you can use to get the circuit working. You can build it on a piece of perf board. If you use this approach, match my component layout as closely as possible, and keep your leads short to prevent problems with stray capacitance. If this is all too much trouble, I supply a two-sided, plated-through board with component placement marked. It comes with or without parts. See the parts list for ordering information. This option will be, by far, the easiest way to get your EPROM programmer working.

If you purchase my bare-board or parts kit, feed-through wire installation is not required, as this board comes with plated-through holes. ALL parts mount on the component side of the board. If you purchase my board, all soldering is done on the "wiring" side of the board.

Step by Step

If you're building the circuit on perf-board, you're on your own; follow the schematic, then proceed to Step 7.

STEP 1 - THE SOCKETS

Install all IC sockets onto the board, making sure that their pin 1 ends (labelled "1" or with a notch) are installed at the end marked "1" at each mounting area. Carefully but completely solder them in place. Use extra care at small pads that have PC traces adjacent to them, as it is easy to get a tiny short

Eprom Programmer Circuit

(c)1984 The John Oliger Co.

The following is available

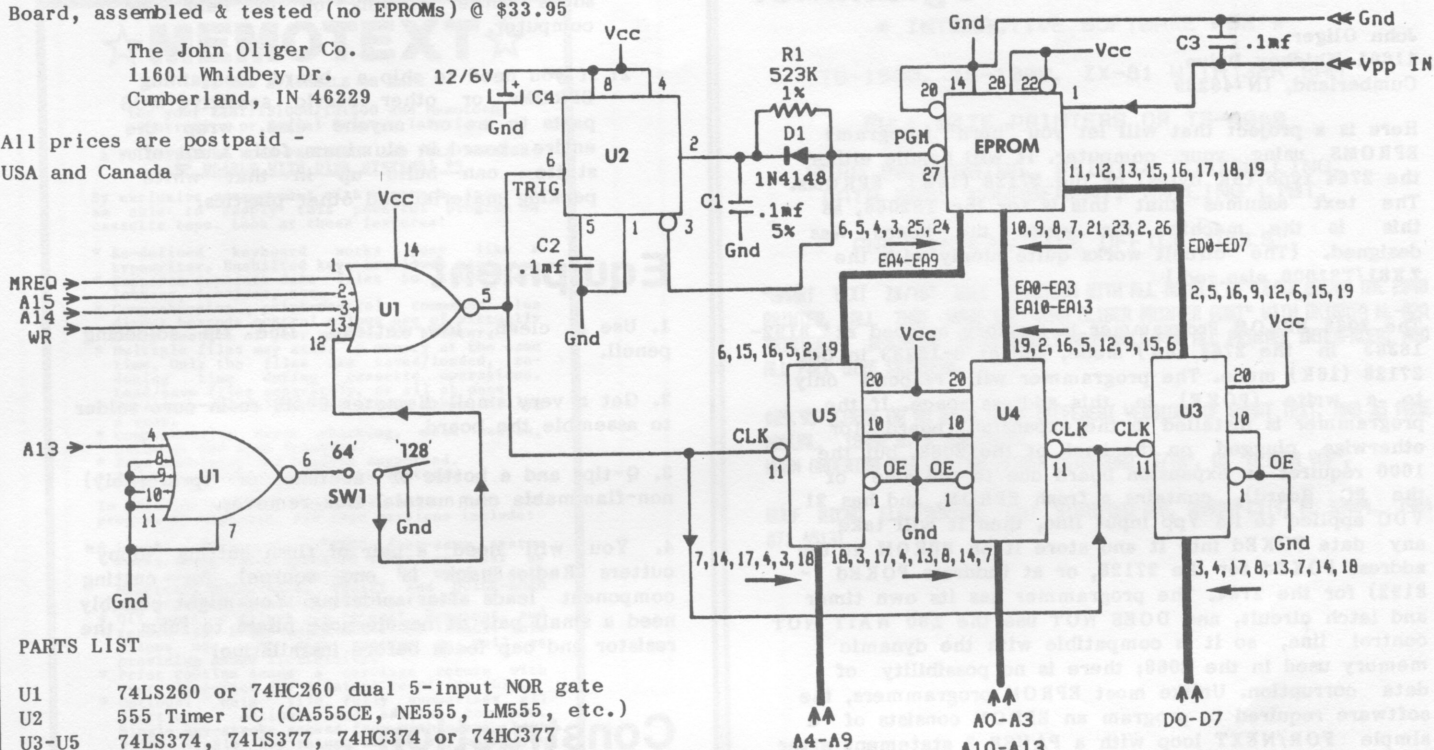
Bare PC board @ \$12.95

PC w/parts (no EPROMs) @ \$25.95

Board, assembled & tested (no EPROMs) @ \$33.95

The John Oliger Co.
11601 Whidbey Dr.
Cumberland, IN 46229

All prices are postpaid -
USA and Canada



PARTS LIST

- ```

U1 74LS260 or 74HC260 dual 5-input NOR gate
U2 555 Timer IC (CA555CE, NE555, LM555, etc.)
U3-U5 74LS374, 74LS377, 74HC374 or 74HC377
C1 .1 uF. mylar capacitor, 5% tolerance or
 better
C2,C3 .1 uF. disc or mylar capacitor
C4 10 uF., 6V. or better tantalum capacitor
R1 523K 1% precision resistor (grn-red-org-org-
 brn) or 1 meg multi-turn pot adjusted with
 DVM
SW1 SPDT ultra-miniature pc mount slide switch
D1 1N4148 switching diode
1 pc 8-pin low profile IC socket

```

- |       |                                                               |
|-------|---------------------------------------------------------------|
| 1 pc  | 14-pin low profile IC socket                                  |
| 3 pcs | 20-pin low profile IC socket                                  |
| 1 pc  | 28-pin low profile IC socket (or user-supplied ZIF IC socket) |
| 1 pc  | 22 AWG by 2" solid red wire                                   |
| 1 pc  | 22 AWG by 2" solid black wire                                 |

at these locations; a little extra care here can save a lot of time in troubleshooting. If you wish to use a ZIF (zero insertion force) socket, it should be installed instead of the low-profile socket provided in my parts kit; it is very hard (to say the least) to remove sockets from a plated-through board once installed. The board is laid out to allow plenty of room for the larger ZIF socket. When the sockets have been soldered in, remove all traces of flux with Q-tips and flux remover or acetone. Carefully inspect your work for solder bridges and bad solder joints, and touch up anything that looks at all suspicious.

## STEP 2 - SMALL PARTS

Install all capacitors and resistors onto the board (C1, C2, C3, C4 and R1). Be sure that C1 is in the correct location, as this capacitor is a low-drift, tighter spec part than the other two 0.1  $\mu$ F. caps in the circuit. (It is the larger 0.1  $\mu$ F. cap in my parts kit.) Also insure that the red (+) end of C4 is installed at the pad marked "+" on the board.

### STEP 3 - SWITCH AND DIODE

Install switch SW1 in its correct location on the board. It should be oriented so as not to cover up too much of the "128" legend etched on the board. Install diode D1, with its cathode (banded) end at the same side as the line on the diode symbol on the board. This results in the diode's banded end being on the left side of the diode as seen looking down at the board with its edge traces toward you.

#### STEP 4 - $V_{pp}$ INPUT WIRES

Prepare 2 pieces of 22 or 24 AWG solid wires (one red, one black), 2 inches long, by stripping  $1/4$ " of insulation from one end of each, and  $3/4$ " from the opposite end. Bend the longer stripped end of each wire into an "O" with small needle-nose pliers. (This has already been done to the wires supplied in the parts kit.) Install these wires on the board by inserting each shorter stripped end into its respective pad on the component side of the board, and soldering them on the opposite side. The red wire goes into the pad marked "Vpp," and the black wire



goes to the pad marked "GND" at the top of the board. The free end of these "Vpp input wires" should be bent to extend above the board (away from the edge traces) for easy attachment to the Vpp supply when programming an EPROM.

#### STEP 5 - CLEAN-UP

Clean off all traces of flux from new soldering, and carefully perform a final inspection. Touch up your work as needed. Soldering on the board is now complete!

#### STEP 6 - INSTALL IC's

Install all IC's (U1-U5) in their respective sockets per the schematic. Because each different type of IC has a different number of pins, it is very hard to get the IC's in the wrong sockets. Be VERY sure, however, that ALL pins of each IC actually enter their respective contacts, and that pin 1 is located at the same end as the "1" label on the board. Beware when installing an EPROM later on, because its pin 1 is on the OPPOSITE end of the socket in relation to U1-U5, usually resulting in its part number printing being upside down when the board is installed. Your programmer is now ready for testing!

#### STEP 7 - THE SMOKE TEST

I recommend testing the board before actually trying to program an EPROM. A logic probe is recommended, but you will be able to do some of the tests even with only a voltmeter. With SW1 in its "64" position, verify that pin 1 of U3, U4 and U5 each receives a pulse when any POKE is made to memory from 8192 to 16383d. Now key in LET a=0 (ENTER). Switch SW1 to its "128" position and key in POKE a,a (ENTER) while monitoring pin 1 of U3. You should receive a pulse on your logic probe. Now use your logic probe or a voltmeter to verify that each used output (see schematic) of U3, U4 and U5 is at a logic low. If not, check the board for errors. Now flip SW1 to its "64" position and enter POKE 16383, 255. Use your logic probe or voltmeter again on all these latch outputs, this time verifying a logic high on each used output. (Don't be confused by a wrong logic level on any outputs that are not used; there are a few such in this circuit.) If you don't get a logic high on every used output, check for errors to find out why. If all is well, your board has passed the "smoke test."

## Power Supply Notes

Although the John Oliger Co., Vpp power supply is recommended for use with this programmer, you can also use your own power supply for Vpp if it is a regulated, variable supply capable of covering the voltage range from 4.4 to 21 VDC. You will also need a voltmeter to use with it that covers this range.

To use it with this programmer, you have to preset it to turn on at 4.5 VDC on power-up. Then, when the instructions say to connect the Vpp supply to the programmer, you connect instead the combination of your preset supply and voltmeter. When the instructions say to plug in the Vpp supply, you should plug in or switch on your own supply, which at this time should be set to 4.5 VDC. When the instructions say to switch the Vpp supply to "Vpp," you should then carefully increase the voltage on your supply to 21 VDC. Use care here not to overshoot 21V, because the EPROM cannot handle anything, even for an instant,

greater than 22 VDC. Any overshoot higher than this WILL destroy your EPROM! (Just connecting the power supply preset to 21 VDC when ready or switching the Vpp input line with a switch will usually destroy an EPROM due to the initial surge!)

When the documentation says to switch the Vpp supply back to 4.4V, you should lower your supply's voltage to about 4.5 VDC. And, of course, when it says to unplug the Vpp supply you should unplug or switch off your supply.

You will find that there is nothing wrong with this arrangement if you only infrequently program EPROMs on this board, but are advised, if for nothing else but convenience, to obtain the Oliger Vpp supply if you find yourself programming EPROMs a lot.

## Burning a 2764 Eprom

When programming 2764 EPROMs, use the "64" position of SW1 and POKE your data to locations 8192-16383. The programmer's base address in this mode is 8192, so a POKE to 8192 goes to the EPROM's location 0, POKEing 8193 goes to the EPROM's location 1, etc. When programing an EPROM on this programmer, the EPROM can be thought of as the famous "write-only memory," which is what it really is at this point!

Install a new or erased 2764 EPROM in the EPROM socket of the programmer, with the computer off and programmer removed from the expansion board. Be sure that pin 1 is installed in the socket at the same end as the "1" etched on the board. Note that the EPROM has its pin 1 to the right, while the support chips on the board all have pin 1 to the left. BE SURE it's plugged in the right way. It will most certainly be destroyed if you try to program it when it is installed backwards! (Although I have not yet mixed up pin 1 on the 2068, I do know that the 1000 will not come up running with the eprom in backwards, let alone be able to program it.-ed.)

Now plug the programmer into your 2068 Expansion Board and attach the Vpp supply's output clips to the programmer board's Vpp input wires.

BE SURE to get polarity right -  
"Gnd" to the supply ground, and  
"Vpp" to the (positive) Vpp output  
of the supply.

Turn on your TS2068 computer. Set the Vpp supply's switches to "4.4" and "21," (There is also a 25V position for programming a 2716 eprom, using a different board.-ed) and plug in the supply. Its green "power on" LED should light, but its red "Vpp" LED should remain off. You are now ready to program your EPROM.

To program a 2764 for use on the 2068 User Cartridge, refer to the program and instructions included with your Cartridge. To program a 2764 with data, such as a machine code program, LOAD or key in your code into free RAM at this time. When the desired data is in RAM, use a FOR/NEXT loop to POKE the data into the EPROM. An example follows:

This subroutine will take the contents of RAM from 32768-40959 (8000-9FFF) and put it on a 2764 EPROM. The programmer's SW1 must be set to "64" or errors will occur in the first few locations. Key in the program with the Vpp supply at 4.4V. Flip your Vpp switch to "Vpp" only right before you GOTO 1.

```

10 LET x=8192 ;Set base addr of pgmr
20 FOR n=32768 TO 40959 ;Data limits
30 POKE x,PEEK n : ;Program 1 byte
 LET x=x+1 and point at next one
40 PAUSE 3 : ;Wait for pgmr to time out
 PRINT AT 9,9;x and print data pgmd.
50 Next n : BEEP 2,2 ;loop back until done
 beep if complete

```

When the TS2068 is done storing the data on EPROM, as shown by the error code or BEEP, switch the Vpp supply back to 4.4 V. Now unplug the Vpp supply and then switch off the computer. Remove the programmer board from the expansion board, and the EPROM from the programmer. Your 2764 is now programmed.

## Program the 27128

On the 2068, programming the 27128 is somewhat more involved than 2764's because the home ROM contains a routine that writes garbage to the first four locations of itself when handling a number from BASIC. This usually results in the EPROM's 4th location (0003h) being programmed a zero, despite attempts at other values. NOTE: This does not happen when programming 2764's with SW1 in the "64" position, so no special programming care is needed with 2764's.

There IS a way around the problem, but care must be exercised nonetheless. An EPROM CANNOT be programmed if the Vpp supply is in its 4.4V position. Also, reading numbers from a DATA list or using a variable that has been set up using a LET statement with Vpp at 4.4V does not cause the ROM to write its garbage. Thus the following two routines have been written to program 27128's on this programmer. The first will take data from RAM locations 32768-49151 and program it on a 27128 EPROM. The second routine replaces the User Cartridge documentation routine when programming 27128 EPROMs with BASIC programs for the User Cartridge.

Important things to remember with 27128's are:

- 1: Keep the 4.4/Vpp switch in the 4.4 Volt position AT ALL TIMES when entering BASIC lines or ANYTHING with numbers from BASIC.
- 2: Switch the Vpp supply to "Vpp" right before you actually start the program. DO NOT start the program with GOTO (number), use GOTO (variable) instead, with the variable equal to the starting line number using a (LET variable = number) statement while the Vpp supply is still in its 4.4 V setting.
- 3: Type in the routine used EXACTLY as listed, except for the numbers listed in the DATA statement as being set for the particular application.
- 4: Be sure SW1 is in its "128" position.

Other than the added care that must be taken, the base address starting at 0, and the 16K capacity of the 27128, programming a 27128 EPROM is just like burning a 2764. I suggest burning only 2764's until you get the feel for the procedure, then adding to your expertise by going on to the 27128.

### Listing 1:

This routine takes the contents of RAM from 32768 to 49151 and programs it onto a 27128 EPROM. This data could be machine code loaded here from tape, another EPROM mapped 32768-49151, or data POKEd into memory in some manner.

```

9990 RESTORE data: READ src: READ end: READ dest:
READ one: READ three: READ loop: READ line: READ col

```

```

9992 POKE dest,PEEK src
9993 PAUSE three
9994 LET src=src+one
9995 LET dest=dest+one
9996 PRINT AT line,col;dest
9997 IF src<=end THEN GO TO loop
9999 DATA 32768,49151,0,1,3,9992,10,13
 (first 2 numbers can be changed if data is
 elsewhere in memory)

```

BEFORE RUNNING THIS SUBROUTINE KEY IN: LET p=9990:  
LET data=9999 (Enter) then flip SW1 to "Vpp" and key  
in GO TO p (Enter).

### Listing 2

This subroutine takes the place of the one given in the 2068 User Cartridge documentation when programming 27128 EPROMs. (PEEK 23627+PEEK 23628\*256)-1 cannot exceed 43085 for one 27128 or 59469 for use with two 27128's. The 2nd data item in line 9999 (xxxx) should be the number written down from the User Cart documentation if 43085 or less. If greater than 43085, use 43085 for the first EPROM, and for the second EPROM use 43086 to replace 26710 in line 9999, and the number written down to replace the xxxx. Note: If the number was greater than 43085 but less than 51278, you can use a 2764 for the 2nd EPROM and save some money. Just use the diodes for the 27128 connected to the 32 and 40 LS138 outputs, and a wire-wrap jumper from the 48 output to the Eb pad on the User Cartridge board.

```

9980 RESTORE data: READ dest: READ one: READ three:
READ eight: READ next: READ data
9982 READ z: POKE dest,z: LET dest=dest+one
9984 PAUSE three
9986 IF dest<eight THEN GO TO next
9988 DATA 0,1,3,8,9982,9999
9989 DATA 1,2,8,128,15,1,0,0
9990 RESTORE data: READ src: READ end: READ loop:
READ line: READ col
9992 POKE dest,PEEK src
9993 PAUSE three
9994 LET src=src+one
9995 LET dest=dest+one
9996 PRINT AT line,col;dest
9997 IF src<=end THEN GO TO loop
9999 DATA 26710,xxxx,9992,10,13

```

TO START THIS SUBROUTINE, KEY IN: LET p=9980: LET  
data=9988 (Enter) then switch Vpp supply to "Vpp"  
position and key in: GO TO p (Enter).





# FAT M-Script

Jack Dohaney  
325 O'Connor St.  
Menlo Park, CA 94025

Use this listing in place of the Basic loader supplied with M-Script. It gives a backup capability and will provide larger letters that help improve readability.

```
1 REM MSCRIPT LOADER WITH FAT
CHARACTERS
3 CLEAR 64736
5 DATA 017,000,253,213,001,00
0,003,042,054,092,036,125,167,03
1,182,018,035,019,013,032,246,01
6,244,225,037,034,054,092,201
7 FOR I=64737 TO 64765: READ
N: POKE I,N: NEXT I
8 RANDOMIZE USA 64737
9 CLS : INK 6: PAPER 0: FLASH
1: PRINT AT 10,6;"LOADING MSCRI
PT CODE": FLASH 0
10 LOAD ""CODE
11 CLS : PRINT AT 20,0;"PRESS
'b' FOR BACK UP","ENTER TO START
MSCRIPT"
12 IF INKEY#="" THEN GO TO 12
13 IF INKEY#="b" OR INKEY#="B"
THEN GO TO 15
14 GO TO 20
15 CLS : PRINT AT 15,0;"MAKING
BACK UP": SAVE "MSCRIPT" LINE 1
: PAUSE 100: SAVE "MSCRIPT"CODE
36864,7931: CLS : PRINT AT 15,0;
"REWIND AND PLAY TO VERIFY": VER
IFY "" : VERIFY ""CODE : PRINT "U
ERIFY OK": PAUSE 200: GO TO 11
20 DELETE 1,19
21 CLEAR 64767: CLS : RANDOMIZ
E USR 36864
```

## \*\*\*\*\* BIG CONTEST!

Last issue SyncWare News announced a contest for programmers. To date, response has been dismal. Not one entry has been received. Now we know that T/S owners are an enthusiastic fun loving bunch. So what gives? Let's get some programs in here!

To refresh your memory, we asked you to send "screen utilities". These can be any program as long as something gets printed on the TV. Judges will be looking for cleverness and imagination. They are not concerned with the language you program in. If you don't know a stitch of machine code, don't worry. You still have a good chance to win a glorious prize (the way things are going you don't even need to know a stitch of Basic to win. All you need to do is enter.)!

So everybody dust off those keyboards and join the fun. Deadline for all submissions is May 15, 1985. YOU could walk away with a fabulous Microdrive, a dazzling Romswitch, a massive Byte-Back 64K ram, or a beautiful John Oliver video upgrade!

Send your entries (as many as you like) on cassette to SyncWare News, P.O. Box 64, Jefferson, NH 03583. Mark your envelope "Display Competition". Show us how you make your micro strut it's stuff! You may be a winner!

**SUPER Prizes,  
Fame, and Glory  
can be  
YOURS!**

\*\*\*\*\*

# SyncWare News

## CONTENTS

|                                     |    |
|-------------------------------------|----|
| For Your Support....                | 2  |
| Forum....                           | 3  |
| Tyd-Byts....                        | 5  |
| Skipper, Teague                     |    |
| TS1000 Plane Frame REVIEW....       | 6  |
| Bent                                |    |
| Clear That Screen (TS1000)....      | 8  |
| Nachbaur                            |    |
| 2068 Word Images....                | 13 |
| Woods                               |    |
| Basil's Compendium (TS1000)....     | 15 |
| Wentworth                           |    |
| 2068 Art....                        | 17 |
| Kulp                                |    |
| Build Your Own Eprom Programmer.... | 19 |
| Oliger                              |    |
| TS1000 Banner....                   | 23 |
| Meyer                               |    |
| FAT Mscript....                     | 24 |
| Dohaney                             |    |

Editor:  
Tom Bent  
9016 Flicker Place  
Columbia, MD 21045

Technical Director:  
Fred Nachbaur  
902 Hoover St  
Nelson, BC Canada V1L-4X6

Advertising, Subscriptions:  
Thomas B. Woods  
P.O. Box 64  
Jefferson, NH 03583

Submissions, announcements, letters to the editor, and all other material of editorial interest should be sent to the Maryland address.

Copyright 1985 SyncWare News

## Subscribe! \$16.95

1 year (6 issues)

Add \$3 a year in Canada; all other foreign add \$5 per year.

FIRST-CLASS MAIL  
U.S. POSTAGE  
PAID  
Jefferson, NH  
Permit No. 1